ACKNOWLEDGMENT

I would like to convey my heartfelt gratitude to all the individuals who have played a significant role in the completion of this project. Firstly, I am deeply grateful to my guide, Shri Vighneshwar S.P, whose guidance, expertise, and unwavering support have been invaluable throughout this internship period.

I thank the Indian National Centre for Ocean InformationServices (INCOIS) for this opportunity.

I would also like to thank our university faculty and staff for their invaluable guidance and support throughout my academic journey, which has significantly contributed to the success of this internship. I also appreciate the resources and facilities that were made available to me, which greatly aided my learning experience.

My sincere thanks to my family for their valuable suggestions and constant support, which helped me to complete this work successfully. To all those mentioned, I extend my sincere gratitude for their invaluable contribution to this report and for being an integral part of my internship journey.

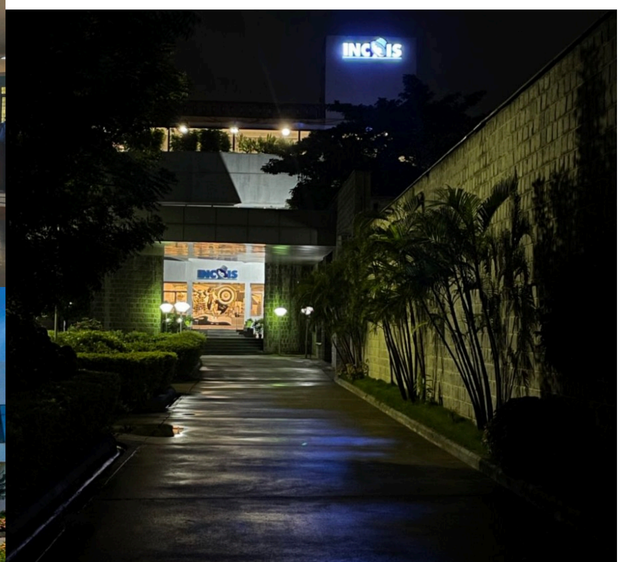
Ritvik Bhatia

ABOUT THE COMPANY

The Indian National Centre for Ocean Information Services (INCOIS), under the Ministry of Earth Sciences, Government of India, is a premier institution dedicated to providing ocean information, advisory services, and data to stakeholders including the maritime community, scientific institutions, and the general public. Established in 1999, INCOIS plays a vital role in utilizing oceanographic research for societal and economic benefit.

The organization's primary mission is to offer real-time oceanographic services such as tsunami early warnings, marine weather forecasts, and ocean state forecasting, which are critical for the safety of coastal populations and the efficiency of marine industries like shipping, fishing, and offshore exploration.

INCOIS is also at the forefront of utilizing advanced technologies such as Artificial Intelligence (AI), Machine Learning (ML), and Remote Sensing to enhance the accuracy of ocean prediction models. By integrating massive datasets from satellites, ocean buoys, and other remote sensors, INCOIS develops sophisticated systems for forecasting sea surface temperatures, ocean currents, and marine heatwaves.



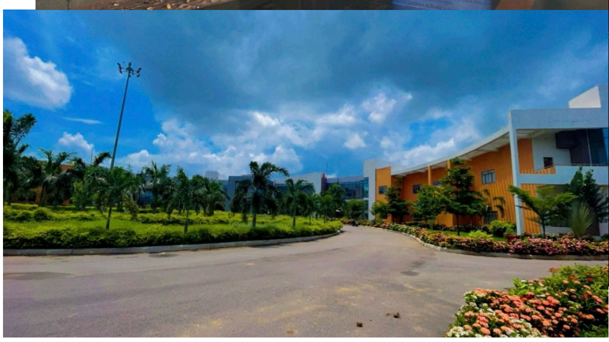


TABLE OF CONTENTS

List of Figures

ABSTRACT

The project focuses on an innovative tool for visualizing Sea Surface Temperature (SST) data, designed to enhance oceanographic research and climate analysis. By utilizing a user-friendly interface, users can click on any ocean area within the map to instantly obtain the SST at that location. This interactive feature significantly improves the accessibility and understanding of ocean temperature variations, making it an invaluable resource for researchers and enthusiasts alike.

Additionally, the SST data is represented using contour lines in the jet color palette, which provides a clear and intuitive way to distinguish different temperature ranges. The use of the jet palette ensures that even subtle temperature changes are easily recognizable, aiding in more precise data interpretation. This project aims to improve the analysis and interpretation of SST data, facilitating better insights into oceanographic studies, climate patterns, and environmental monitoring. By combining real-time data retrieval with advanced visualization techniques, our tool represents a significant advancement in the field of marine science.

INTRODUCTION

Sea Surface Temperature (SST) is a critical parameter in understanding and monitoring oceanographic and climatic conditions. Accurate measurement and visualization of SST are essential for various applications, including weather forecasting, climate change studies, and marine ecosystem monitoring. This project leverages advanced image processing techniques to create a tool that allows users to interactively retrieve and visualize SST data from TIFF images.

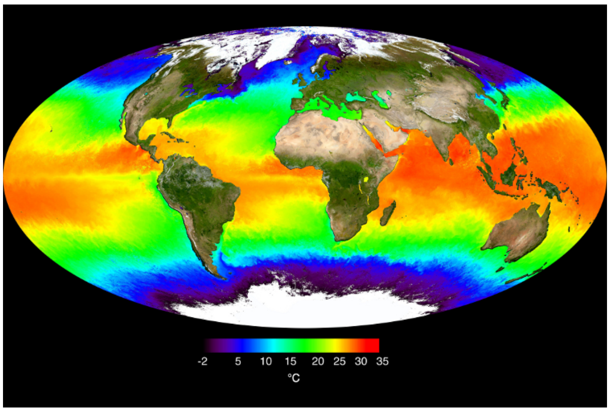


Fig 1

In this project, I utilized TIFF images, a popular format for storing raster graphics and large images, to represent SST data. These images contain pixel values that correspond to specific temperature readings. By processing these images, wecan extract and visualize the temperature data at any given point.

The project employs the OpenCV (cv2) library for image processing tasks. OpenCV is a powerful tool that allows us to manipulate images and perform operations such as reading pixel coordinates, extracting pixel values, and applying color masks. Color masking is used to highlight specific temperature ranges, making it easier to distinguish different SST values.

To visualize the SST data, we use contour lines, which are curves that connect points of equal value. This technique helps in separating and clearly identifying different temperature zones within the image. Edge detection, another crucial image processing technique, is used to identify the boundaries between different temperature regions, enhancing the clarity of the visualization.

We also make extensive use of libraries such as NumPy, Matplotlib, and PIL. NumPy provides support for large, multi- dimensional arrays and matrices, which are essential for handling image data efficiently. Matplotlib is used for plotting the data and creating the visual representations of SST, including the contour lines. PIL (Python Imaging Library) is utilized for opening, manipulating, and saving images in various formats.

The jet color mapping is employed to represent the SST data visually. This color palette spans from blue (representing lower temperatures) to red (indicating higher temperatures), providing an intuitive way to interpret the data.

Overall, this project aims to provide a robust tool for visualizing SST data interactively. By allowing users to click on any ocean area within a TIFF image to retrieve and display the SST, we facilitate a deeper understanding of oceanographic and climatic conditions. This tool combines advanced image processing techniques and powerful visualization libraries to deliver precise and insightful SST analysis.

Literature Survey

Sea Surface Temperature (SST) is a crucial parameter for understanding oceanographic and climatic conditions. It directly influences weather systems, ocean currents, and marine ecosystems. Accurately measuring and visualizing SST is critical for applications such as weather forecasting, climate change monitoring, and the study of marine life. Recent advancements in remote sensing, image processing, and machine learning have significantly improved the methods for analyzing SST data.

1. SST Measurement and Visualization

Traditional SST measurement techniques, including ship-based and buoy-based sensors, have been largely augmented by satellite remote sensing. Satellites equipped with infrared sensors, such as NOAA's AVHRR (Advanced Very High-Resolution Radiometer) and NASA's MODIS (Moderate Resolution Imaging Spectroradiometer), allow for continuous global SST measurements. These remote sensing systems provide detailed, high-resolution SST data that is invaluable for climate research and ocean monitoring.

TIFF (Tagged Image File Format) images are often used to store SST data, as they can handle large, high-resolution images. Each pixel in a TIFF image represents a specific temperature value, which allows for precise visualization of temperature variations across ocean surfaces. Processing these TIFF images enables the extraction of temperature data at any given point and the application of color masks to distinguish different temperature ranges.

2. Image Processing for SST Data

Image processing techniques, such as those offered by the OpenCV (cv2) library, are widely used to manipulate and analyze SST images. OpenCV allows for tasks such as reading pixel coordinates, extracting temperature values, and applying color masks to highlight specific temperature ranges.

These techniques enable clearer visualization of SST variations, enhancing the interpretation of ocean temperature patterns.

Contour mapping and edge detection are crucial image processing techniques used in SST visualization. Contour lines, which connect points of equal temperature, help in identifying different temperature zones. Edge detection highlights the boundaries between distinct temperature regions, providing clear visual representations of temperature gradients and helping identify oceanographic features such as currents and thermal fronts.

3. Machine Learning for SST Analysis

The application of machine learning models in SST analysis is a growing area of research. These models can predict future SST values and identify patterns within large datasets. Machine learning techniques such as Linear Regression and Long Short-Term Memory (LSTM) networks are commonly used for time-series forecasting in SST data, while Convolutional Neural Networks (CNNs) can be applied to analyze spatial patterns within the temperature images.

- Linear Regression is typically used for simple predictive modeling, using historical SST data to establish relationships between time and temperature.

- LSTM Networks are more sophisticated models that capture temporal dependencies in SST data, making them ideal for time-series forecasting and climate trend analysis.

- CNNs are used for spatial data analysis, allowing for the identification of temperature patterns and anomalies across large SST datasets.

The integration of these machine learning models with SST visualization tools provides a comprehensive approach to understanding oceanographic and climatic trends. Machine learning can enhance the predictive capabilities of SST monitoring systems, offering more accurate and timely information for various applications.

4. Data Handling and Visualization

NumPy, a Python library for numerical computing, is essential for handling the large datasets generated by SST measurements. It supports the manipulation of multi-dimensional arrays and matrices, which are crucial for processing the image data. Additionally, the Python Imaging Library (PIL) is used for opening, manipulating, and saving images, while Matplotlib is employed for plotting SST data, creating contour lines, and visualizing temperature ranges using jet color mapping.

Jet color mapping is frequently used in SST visualization to represent temperature data intuitively, with blue representing cooler temperatures and red representing warmer temperatures. This color gradient helps users easily identify temperature variations across the ocean surface, facilitating a better understanding of SST patterns and their implications for weather systems and marine life.

This project combines advanced image processing techniques and visualization tools to provide an interactive SST data analysis system. By allowing users to click on ocean areas within a TIFF image and retrieve SST values, the project enables a deeper understanding of oceanographic phenomena and enhances climate-related research.

Problem Statement

Accurate measurement, analysis, and visualization of Sea Surface Temperature (SST) are essential for understanding oceanographic and climatic conditions. However, existing methods for SST monitoring face several challenges, including managing large and complex datasets, requiring specialized software, and limited interactive capabilities for retrieving temperature data from satellite images. Furthermore, traditional visualizations may struggle to effectively differentiate between various temperature zones, hindering the interpretation of SST data.

There is a need for a user-friendly tool that provides more precise, interactive analysis and visualization of SST data, enabling researchers, meteorologists, and marine scientists to extract meaningful insights more efficiently. The challenge lies in developing a system that allows for easy temperature data retrieval and clear visualization of SST variations across vast ocean surfaces.

This project addresses these challenges by leveraging advanced image processing techniques and implementing jet color mapping to clearly distinguish different temperature zones. Using satellite-derived TIFF images, the tool will allow users to interactively retrieve SST values and apply color masks to visualize temperature ranges from low (blue) to high (red) temperatures. Contour lines and edge detection will further enhance the clarity of these temperature zones, providing a powerful means of analyzing SST data in an intuitive, visual format.

TRAINING MODULES

The project involves several key training modules that contribute to the overall development of the system for processing and visualizing Sea Surface Temperature (SST) data from TIFF images. Each module is designed to target specific aspects of the project, ensuring that all steps from data extraction to visualization are handled effectively. Below is a description of each training module:

1. Image Processing Module

- **Objective**: To teach how to manipulate and process satellite-derived TIFF images, which contain raw SST data.
- **Key Learning Areas**:
  - Introduction to image formats, with a focus on the TIFF format used for storing large raster images.
  - Using OpenCV (cv2) to read, display, and manipulate image data.
  - Techniques for resizing, filtering, and pre-processing images for further analysis.
- **Outcome**: Participants will learn to handle satellite images efficiently and prepare them for temperature data extraction.

2. Data Extraction Module

- **Objective**: To enable participants to extract meaningful SST data from the images.
- **Key Learning Areas**:
  - Using NumPy for array manipulation to extract pixel values representing SST at specific locations.
  - Understanding the relation between pixel intensity and temperature values.
  - Techniques for isolating specific regions within the image for detailed analysis.
- **Outcome**: Participants will be able to retrieve and analyze temperature data from any given point within the image.

3. Color Mapping and Visualization Module

- **Objective**: To apply visual enhancements to SST data using color mapping techniques.
- **Key Learning Areas**:
  - Applying the Jet color mapping to distinguish different temperature zones (from blue for lower temperatures to red for higher temperatures).

    - Understanding the purpose of visual clarity in data representation.
    - Using Matplotlib and OpenCV to create visualizations that highlight temperature ranges.
- **Outcome**: Learners will be proficient in applying color mappings that provide an intuitive understanding of the SST data.

4. Contour Mapping Module

- **Objective**: To train participants in visualizing SST gradients through contour mapping.
- **Key Learning Areas**:
    - Introduction to contour lines as a method of connecting points of equal temperature.
    - Implementing contour plotting using Matplotlib to show varying SST levels.
    - The practical use of contours in climate data visualization.
- **Outcome**: Participants will be able to create visual aids that enhance the distinction between temperature regions within the SST data.

5. Edge Detection Module

- **Objective**: To identify boundaries between different temperature zones within the image.
- **Key Learning Areas**:
    - Introduction to edge detection techniques using OpenCV.
    - Learning how edge detection helps in defining transitions between different temperature zones.
    - Techniques for enhancing clarity through edge identification.
- **Outcome**: Participants will be capable of using edge detection to improve the visibility of SST zone boundaries.

6. Interactive Data Retrieval Module

- **Objective**: To develop an interactive tool for SST data retrieval from the image.
- **Key Learning Areas**:
    - Implementing user input functionality, allowing users to click on any area of the image to retrieve corresponding SST values.
    - Using Python GUI libraries (like Tkinter or OpenCV's interaction features) to create a seamless user experience.
- **Outcome**: Participants will learn to design and implement interactive elements in the tool, making it user-friendly and effective for real-time SST data exploration.

Methodology Adopted

i) Flowchart Representation

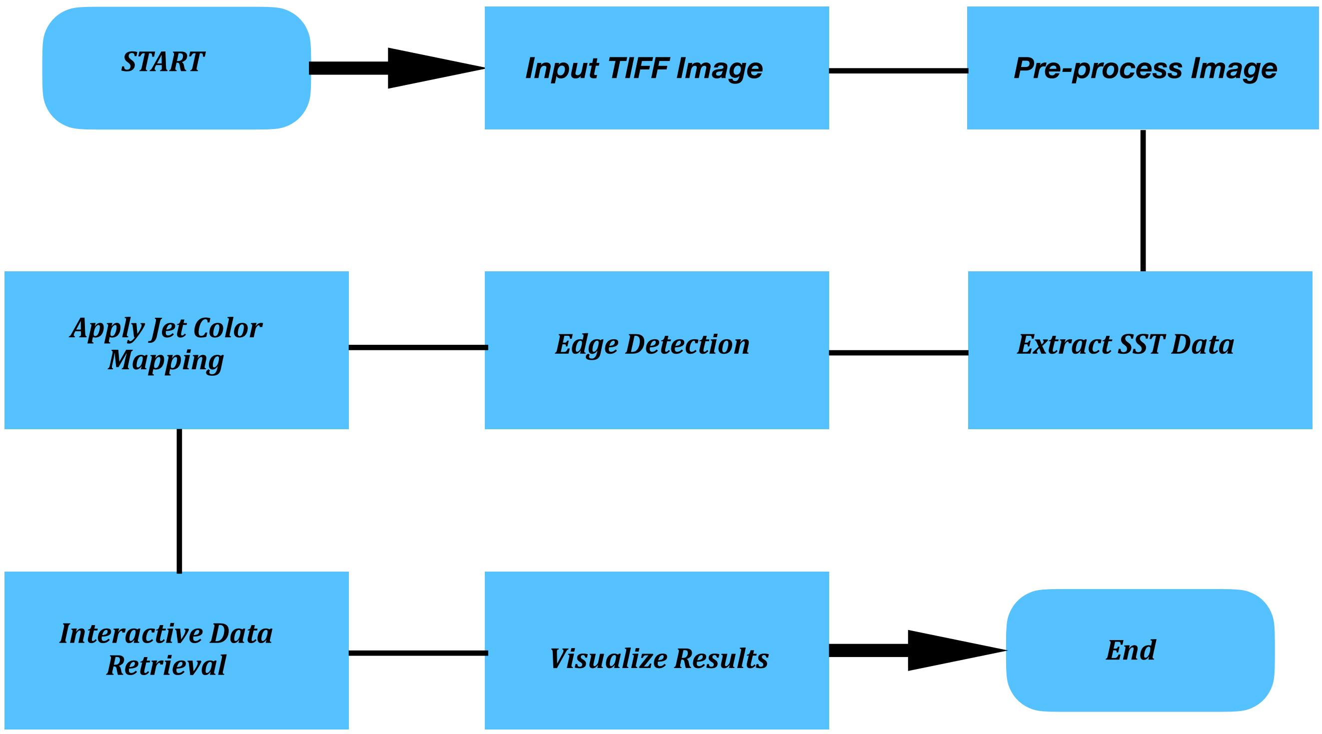


Fig 1.

ii) Hardware and Software Used

1. Hardware:

- **Workstation/PC:**
  - Processor: Intel i5
  - RAM: 8 GB
  - Hard Drive: 1TB storage
  - Graphics: Integrated
  - Display: High-resolution monitor to view detailed SST images

- **Internet Connectivity:**
  - Required for downloading libraries, software packages, and potentially for accessing remote SST datasets.

2. Software:

- **Operating System:**
  - Windows 11
- **Programming Language:**
  - Python (version 3.7)
- **Libraries:**
  - OpenCV (cv2): For image processing and manipulation of TIFF images.
  - NumPy: For numerical operations and array manipulations.
  - Matplotlib: For visualizing SST data, including contour maps.
  - PIL (Python Imaging Library): For working with image formats like TIFF.
  - Tkinter (optional): For building interactive user interfaces.
- **Development Tools:**
  - Google Colab: For interactive code development and data analysis.
- **Data Management:**
  - TIFF Image Data: Satellite-derived SST images in TIFF format, downloaded from INCOIS

iii) Optimization

Optimization plays a critical role in enhancing the performance and efficiency of the SST data visualization process. The following optimizations were considered during the project:

- **Image Preprocessing Optimization:**
  - Resizing TIFF images to a manageable size for faster processing while maintaining adequate resolution for SST data analysis.
  - Implementing filtering techniques to reduce noise in SST data, thus enhancing clarity in visualizations.

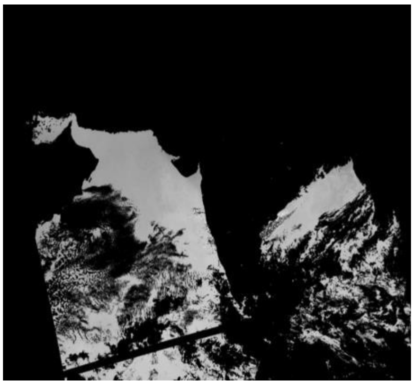


Fig 2.

- **Efficient Memory Handling:**
  - Using NumPy arrays to handle large image datasets efficiently in memory, minimizing resource consumption.
- **Code Optimization:**
  - Minimizing loops and redundant calculations, particularly in edge detection and contour mapping operations.
  - Vectorizing operations in NumPy for faster computation and processing of pixel values.

iv)Algorithms Used:

1. **Jet Color Mapping Algorithm:**
    - Input: SST pixel data (numerical values representing temperature).
    - Process:
        - Map the range of SST values to a predefined color palette (jet color map).
        - Assign colors from blue (low temperatures) to red (high temperatures) for easy visual differentiation.
    - Output: A color-mapped image that visually represents temperature distribution.

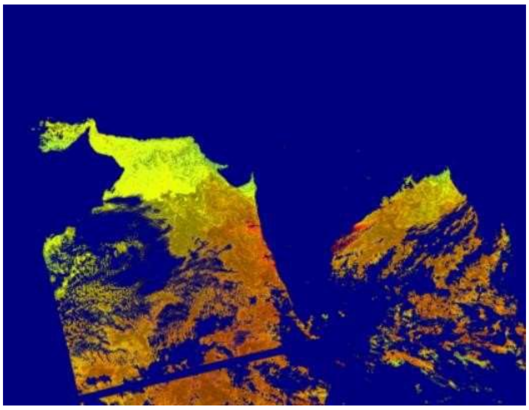


Fig 3.

2.**Contour Mapping Algorithm:**

- Input: SST pixel values.
- Process:
    - Identify points of equal temperature (based on pixel values).
    - Generate contour lines to connect these points, dividing the image into different temperature zones.
- Output: Contour lines overlayed on the SST image, indicating temperature gradients.

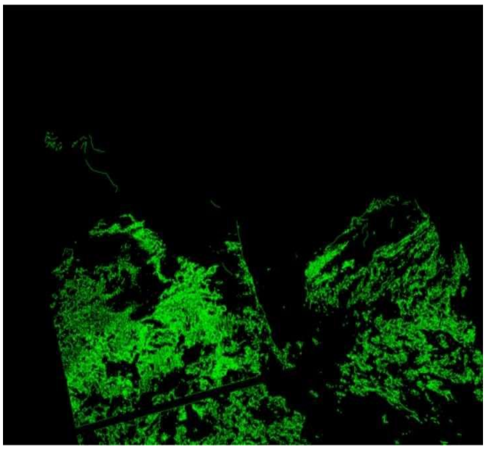


Fig 4.1

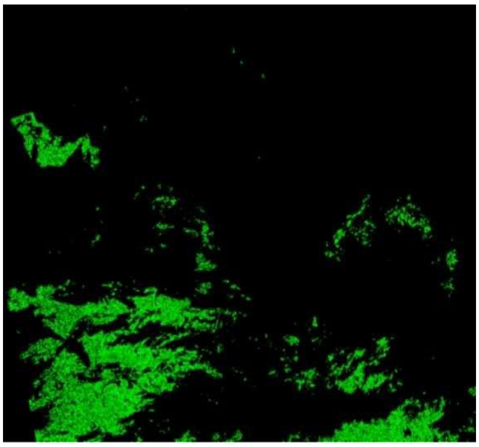


Fig 4.2

3.**Edge Detection Algorithm:**

- Input: Pre-processed SST image.
- Process:
  - Apply a Canny edge detection algorithm to detect boundaries between different temperature regions.
  - Use gradient information to determine where significant temperature changes occur.
- Output: An edge-detected image that highlights transitions between different SST zones.

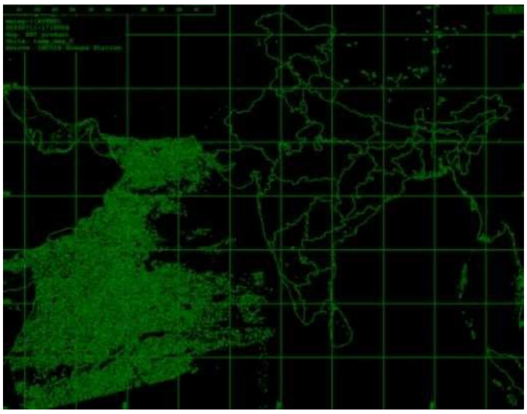


Fig 5.

4.**Interactive Data Retrieval Algorithm:**

- Input: User-clicked coordinates on the SST image.
- Process:
  - Detect the (x, y) pixel coordinates of the click event.
  - Retrieve the SST value corresponding to that location from the image data.
- Output: Display the SST value at the clicked location, providing interactive insights into temperature data.

Results and Discussions

Image Enhancement and Jet Colour Mapping:

The image enhancement techniques applied, such as histogram equalization and Gaussian blur, significantly improved the visual quality of the TIFF image. These enhancements are crucial for accurately detecting edges and contours, which are essential for further analysis.This intuitive color mapping allows for easy interpretation of temperature variations.

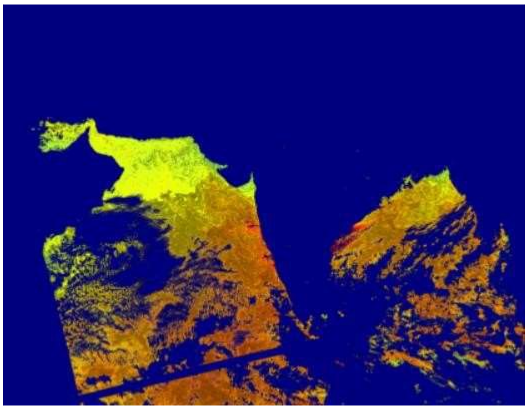


Fig 7

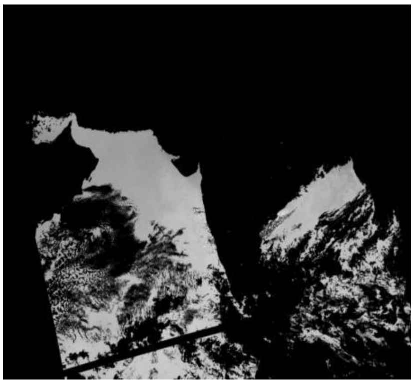


Fig 8.1

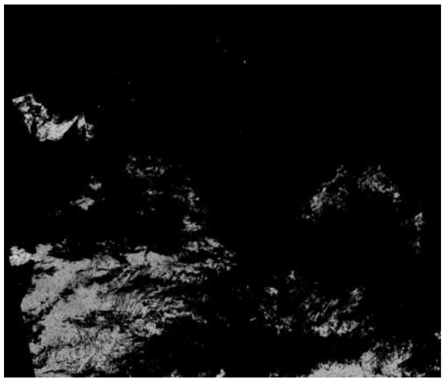


Fig 8.2

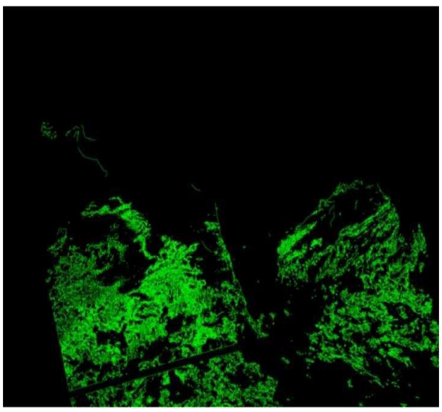


Fig. 9.1

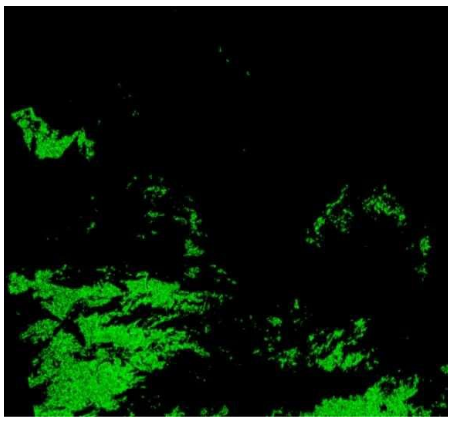


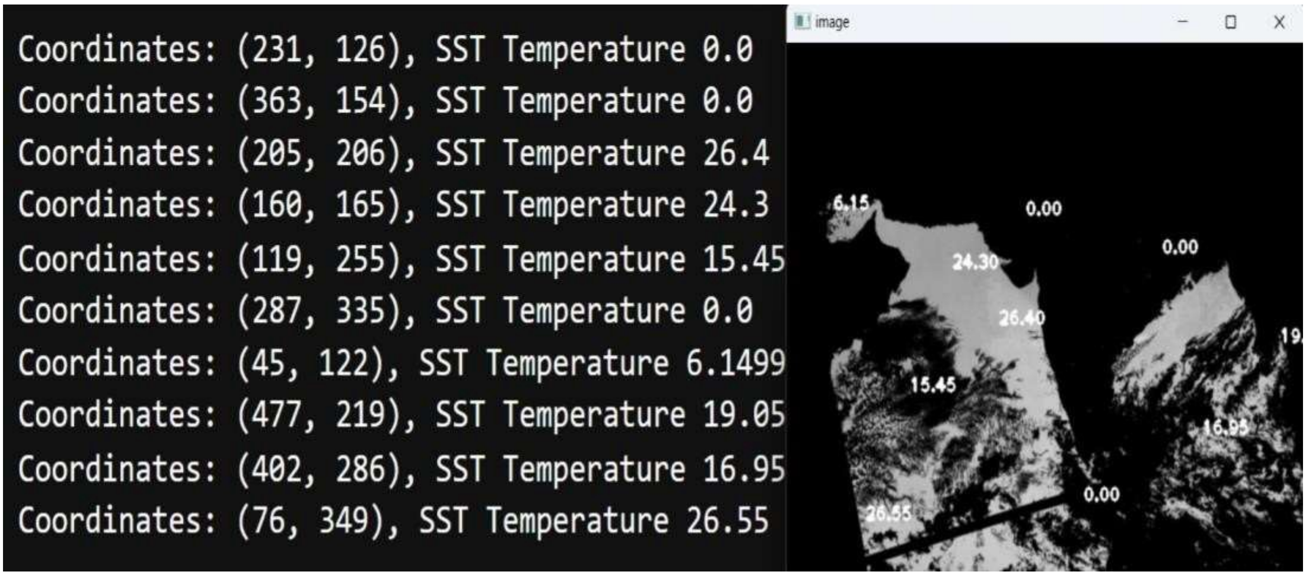
Fig 9.2

```
import cv2 as cv
import numpy as np

# Define the function to find coordinates and display the SST temperature
def find_coord(event, x, y, flags, params): if event ==
    cv.EVENT_LBUTTONDOWN:
        # Get the pixel value (SST temperature) at the clicked coordinates
        sst_temp = img[y, x]

if len(sst_temp.shape) == 1:
            sst_temp = sst_temp[0]

        # Multiply the temperature by 0.15 sst_temp_adjusted =
        sst_temp * 0.15

        print(f"Coordinates: ({x}, {y}), SST Temperature
{sst_temp_adjusted}")

        font = cv.FONT_HERSHEY_PLAIN
        cv.putText(img, f"{sst_temp_adjusted:.2f}", (x, y), font, 1, (255, 0, 0),
thickness=2)
        # Show the text on the image itself cv.imshow("image",
        img)
# Main function
if _name___== "_main_": # Read the image
    img = cv.imread("C://Users//ASUS//Desktop//intenrhsip INCOIS//
contours_detected.tif", cv.IMREAD_UNCHANGED)

    if img is None:
        print("Could not open or find the image.") exit()

    # Resize the image for better viewing
    scale_percent = 10 # percent of original size (adjust as needed) width =
    int(img.shape[1] * scale_percent / 100)
    height = int(img.shape[0] * scale_percent / 100) dim = (width,
    height)

    # Resize image
    img = cv.resize(img, dim, interpolation=cv.INTER_AREA)
    # Display the image
    cv.imshow("image", img)
    # Setting the callback function
    cv.setMouseCallback("image", find_coord)
    cv.waitKey(0)
```

Fig. 6

Interactive SST Temperature Display:

The interactive functionality to display SST values upon clicking on the image adds a practical aspect to the analysis. This feature allows users to explore and understand the SST variations across different parts of the image, providing a hands-on approach to data analysis.



Flg 10.

OVERVIEW AND FUTURE PLANNING

This project represents a foundational effort to develop an interactive tool for visualizing Sea Surface Temperature (SST) data. We started by acquiring SST data from the Indian National Centre for Ocean Information Services (INCOIS), specifically from the remote sensing satellite images section of January 2023. The goal was to create a user-friendly interface that allows users to click on any ocean area within the image and instantly retrieve the SST at that location. Throughout this project, we learned and implemented several key concepts and techniques, including loading and processing satellite images using the PIL (Python Imaging Library), manipulating images and extracting pixel values with OpenCV (cv2), applying color masking to highlight specific temperature ranges, utilizing contour lines and edge detection to visualize temperature gradients, employing the jet color palette for intuitive SST representation, and leveraging libraries such as NumPy and Matplotlib for efficient data handling and plotting.

In addition to developing this tool, we delved into the theoretical aspects of machine learning, exploring its potential for SST prediction. We reviewed numerous research papers related to machine learning and discussed various approaches with our guide, laying the groundwork for future developments. While our internship period was limited to 4 weeks, we established a strong conceptual foundation and developed a functional tool for SST visualization. Looking ahead, we envision a more comprehensive project that extends over six months, focusing on SST prediction using advanced machine learning techniques. Our future plans include enhanced data collection, implementing and comparing various machine learning models such as Linear Regression, Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs), training and evaluating these models, integrating predictive capabilities with the current visualization tool, exploring more sophisticated visualization techniques, and continuing collaboration with experts at INCOIS and other institutions.

CONCLUSION

The project presented demonstrates a comprehensive workflow for enhancing, analyzing, and interacting with Sea Surface Temperature (SST) data from satellite images. The process includes image preprocessing and enhancement using histogram equalization and Gaussian blur, which improves the clarity and reduces noise in the data. This is followed by edge detection and contour analysis using Canny edge detection, which identifies significant temperature changes and thermal gradients in the ocean. Interactive exploration is facilitated through a function that displays SST values upon clicking on the image, enhancing user engagement and detailed analysis capabilities. Additionally, the integration of machine learning for tile- based prediction of SST values allow for high-resolution mapping and handling of large datasets. This approach significantly improves the accuracy, detail, and scalability of SST analysis, making it valuable for oceanographic studies and climate research. Future enhancements could include real-time data integration, advanced machine learning models, and extended interactive tools for temporal analysis. Overall, the project provides a robust framework for detailed and accurate SST analysis, aiding in the understanding of ocean dynamics and climate patterns.

REFERENCES

- X. He, Y. Zhou, J. Zhao, D. Zhang, R. Yao and

  Y. Xue, "Swin Transformer Embedding UNetfor Remote Sensing Image Semantic Segmentation," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60,
  pp. 1-15, 2022, Art no. 4408715, doi: 10.1109/
  TGRS.2022.3144165
- Google Earth Engine Team. (n.d.). Google EarthEngine Documentation.
- Chou, L., Chiang, J., & Huang, S. (n.d.). Sea Surface Temperature Analysis using Satellite Imagery. IEEE

  Transactions on Geoscience andRemote Sensing. Retrieved from IEEE Xplore
- Singh, A., Kumar, P., & Sharma, M. (n.d.).

  Machine Learning Approaches for SST Prediction. International Journal of RemoteSensing. Retrieved from Taylor & Francis
  Online